

DOWN UNDER CLUB

Editor Harry Hungins 12 Thomas Str. Mitchee. 3132 03-873-1408

Treasurer Ron Allen 2 Orlando Str. Hampton, 3188 03-598-4534

A MERRY CHRISTMAS TO ALL OUR MEMBERS.

In the mailbag this week was a pair of 'P' plates, sent by the Editor of our 'Competitor'.

Now why 'P' plates. Have they taken them off their 'PREMIUM' state, or is it a hint that VZDU should drive on 'P' plates? Cheeky So & SOs

The intention was neither. In fact they make a very good packing when sending disks through the post. I hasten to add though, that if you use them, make sure they are the old ones and <u>NOT</u> the newer magnetic type, or you may be sending a blank disk.

But I am displaying them as PROVISIONAL.

Fact is that I have had a stroke and am looking for a new Editor. (See the Trading Post for details of this lucid position. A touch of insanity would also be a help in having your application accepted.) After 4 years I think that someone else should take up the torch. I suggest someone younger. Much younger. If someone cares to take over I will give them all help for the first few issues, and all the data we hold and programs to carry on with. They will also have first option to purchase all or any of my equipment. There are sufficient funds to carry on, which we will transfer to them, unless Ron cares to carry on as treasurer. That is not for me to say.

I shall put out one more issue at least. Jan/Feb. After that depends on circumstances. So the "Ball is in your corner". There is too much "Let George (read Harry) do it" Well the old grey mare ain't what she used to be. I have enjoyed doing this, but of late support has fallen off to zero, and I don't even knowif the rag gets read!

Failing someone taking over I have made a suggestion to HVVZUG that we combine the 2 clubs, or the 2 N/Letters. I shall be in Newcastle over Xmas and will discuss it with Joe. To put your minds at rest, if we do close up, there are sufficient funds to refund everyone their outstanding credits, So you won't be ripped off, as has happened in times past.

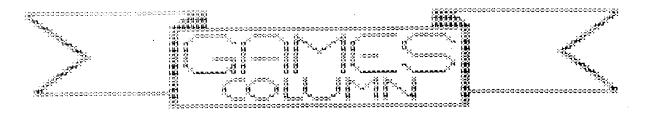
Let us know what you think and any or all suggestions you may have. I will put out one more issue. It is your club so now take over and do something about it. I am sorry. But I can't go on pulling articles out of either my hat or from Bob. Even he must eventually run out.

Now turn to the trading post and write your application.

FOR SALE

VZ300 w/16K mem.expan. DISKDERIVE and CONTROLLER PRINTER INTERFACE. ALL PLUG PACKS AND LEADS. \$250.

PRINTER INTERFACE. ALL PLUG PACKS AND LEADS. \$250.
VZ300 WITH EXPANDED ROM CHIP FITTED DISK DRIVE DISK DR. CONTROLLER WITH EXPANDED DOS CHIP FITTED PRINTER INTERFACE AND ALL LEADS AND POWER SUPPLIES. \$350.
VZ300 COMPUTER\$50.
MEMORY 64K\$75
MEMORY 200 16K\$25
MEMORY 300 16K\$35
PRINTER INTERFACE NEW\$30. USED \$20
WORDPROCESSOR CARTRIDGE\$50
DISK CONTROLLER FITTED WITH EXT. DOS CHIP AND RESET.\$50
DISK DRIVE\$70.
DISK DRIVE POWER SUPPLY\$40.
VZ300. NOT WORKING. 4only. \$20 ea.
LIGHT PEN\$50
JOYSTICKS\$35 PR.
PLUG PACK 97. (2V)-\$15.
CASSETTE RECORDER DR20\$25 " DES 7344- new\$30.
DISK STORAGE BOXES\$10.
BACK ISSUES VZDU. \$1.50 EA POST FREE
VZ ORIGINAL GAMES TAPES. \$3 ea.
PRINTER SEIKOSHA GP250X. Tractor feed\$75.



Welcome to the Christmas issue of the games column.

I've just finished my yr 11 exams and already I have assessments to complete before week something-or-other. Well, lacking any sort of energy and will power I've let the games hints and tips slip from my mind. But I did have a games review prepared for a few issues ago which did not make it to the final drawing board.

So for this issue I will review a game that I'm sure everyone has heard of; "DAWN PATROL."

It is in binary and I can not enter into the program to identify the author. It is very popular, as anyone who has followed the high scores from the begining would know.

You are given the task of rescuing 80 prisoners from four prison camps behind enemy lines. Using the only available means of transport, you fly a helichopter to each camp, land inside the enclosure and wait till all the prisoners, or as many as you can carry, run out to the chopter. But these's a few catches, the more prisoners you collect the more petrol you use up. Camps one to four are guarded be tanks. Tanks can only shoot parrallel to the ground and can get you while you are loading the prisoners. Camps one to three are also guarded by missiles launches, which shoot missiles into the air and travel at 4 degrees left or right. Camps one and two also have 'planes that appear, fire a missile and disappear into the distance. You can not shoot the 'plane but you can shoot the missile. A few times more then a normal surface to air missile.

On top of that your mission starts at 4.00am and terminates at 6.00 am.

You score points for the amount of prisoners you return to base with. Prisoners are worth more depending on which camps they came from. Camp four (the closest) prisoners are worth 100, camp three are worth 200, camp two are worth 500 and camp one are worth 1500.

O.K so it's a pretty short review. How about some-one, ANYONE, giving me some fill-in material. You know ... an opportunity to get your name up in "lights." A drawing or design would do, using one of the drawing programs from our library. It could be a games title, or your version of a little gobbling games monster, or whatever catches your imagination. But it would be a help for fill-in material!

Library tape two has a drawing program called "Sketches". It will print out your drawing. The others won't print, but it is easy to modify them so that they will. Load the program you want, <LIST> it, and find the string commands, then add the print or copy command. Just take lines 3020-3050 in "Sketches" as a guide. These lines contain the string commands. Line 3030 contains the string which will enable you to print your design. If "P" has already been used for something else then just use another letter. Make a note WHICH letter, because if you hit the wrong one you will lose your drawing

and have to start ALL OVER AGAIN.

Don't forget to load "Printer Patch" before you load the drawing program.

Sorry to hear you haven't been well, Harry. But it's the timing of your misfortune that has me worried as well. Just a week beforehand my younger brother Mitch played soccer at South Melbourne. He was delighted to be able to meet Harry at the soccer grounds. THEN, one week later ... oh, dear! As an older brother I have always known of the CURSE of younger brothers!!!

Just kidding everybody! HONEST!!!

Anyway Harry, here's hoping you're feeling great again now and that you and all VZ-ers have a tremendous Christmas and a safe and happy New Year.

TIM.

TRADING POST.



SITUATION SOON TO BECOME VACANT.

Editor of Leading V.Z. Newsletter and Secretary of old established Computer Club.

Any 12 year old person, with Master's Degree in Electronics, Computer Languages, Psychology and Psychiatry, and with the patience of Job, the foresight of Jacob, the manipulating ability of Mandrake and the luck of James Bond would be acceptable.

However, greater Age and lesser qualifications would be considered.

Location; Anywhere Hours; Enormous

Salary: Nil

Any applications to President, care of this Rag.

INTRODUCTION TO PROGRAMMING - PART I

by Bob Kitch

I have been asked to contribute a series on BASIC programming for the VZ-Users. So here goes.

Firstly, the series will be unconventional. Most introductions to BASIC proceed blow-by-blow through the various BASIC commands. I will not - many texts

exist which can explain these better than I can.

Secondly, the series will initially be non-specific to any particular computer language. General programming concepts and guidelines will be offered. The principles will be equally applicable to BASIC, Assembler, Pascal or whatever.

Thirdly, advanced programming concepts and hints will be offered as they are needed. This is the best time to introduce these since their mystique is removed.

Fourthly, early emphasis will be on PLANNING, ORGANISING and MAINTAINING a program, rather than encouraging feverish coding at the keyboard (which is usually commenced too early by beginners).

It is quite possible to recognise a breed of compulsive programmers, born from the home micro boom. This breed, is emerging from the brave new world of tomorrow's technology whose reason for existence is simply to program. People become totally fascinated by the unlimited abstract world that the inside of a computer offers. We can create a Universe or any World inside a machine. In the abstract world of programming, a well thought out programming method serves as a MAP, and the techniques of software engineering are the WEAPONS. These then are the main threads of this series.

Let's commence this month with a few definitions and concepts to ponder over until the next installment.

THE COMPUTER is a machine, and is only capable of doing simple work. It has been termed by some as "a remarkably efficient counting machine with a large memory — but no brains!" It has no intelligence and cannot think.

A COMPUTER SYSTEM consists of four elements :-

- 1. the Central Processor Unit (in the VZ it is the Z-80A microprocessor chip) with "primary memory" (ROM and up to 34K RAM).
 - 2. Input devices keyboard, cassette, disk and so on.
- 3. Output devices screen, printer, cassette, disk, in—built speaker, voice and sound synthesisers etc.

4. "Secondary memory" – not essential but may be cassette or disk when used to update or relieve primary memory.

MAN-MACHINE INTERFACE. The interaction between man-machine inputs and outputs is a continuous and circular feedback process. e.g. man output (keypress) is machine input..or.. machine output (screen prompt) is man input-..and so on. This interaction forms the basis of using computers.

The four fold subdivision of a computer system is little different to our own mental capabilities. The CPU and primary memory is broadly equivalent to our mind. The I/O devices are similar to our senses (touch, taste, sight, sense of heat, speaking, hearing). The secondary memory is directly comparable to our use of external aids to assist our memory, such as note books, filing cabinets of information, telephone directories — all of which have slow access and are difficult to recall compared with things already resident in our mind.

COMPUTER PROCESSES or CAPABILITIES are surprisingly few in number. There are only FOUR and unless an exercise or problem can be broken down into these elementary processes, then coding of the program should not commence. A greater understanding of the problem is required before proceeding.

It is important to clearly distinguish two things whilst programming. The first, is to devise a LOGICAL solution to the programming exercise, which is

quite independent of the particular language to be used. The second, is the actual CODING of the exercise being undertaken. The latter stage is easy, provided that the former is well understood. The computer program will only function correctly if the logic of the program is correct, and there are no aids or diagnostics available from the machine to assist in achieving correctness in this demanding aspect of program design. Some diagnostics are however available to assist in the coding portion of the task – such as the SYNTAX checking.

As one becomes more familiar with programming languages it is soon apparent that many of the powerful command structures are simply macro instructions formed from these few "primatives".

The four processes are :-

- 1. Input data and store it in primary memory the data may be either "raw" data input (e.g. from keyboard) or read—in from the secondary store. (e.g.tape).
- 2. Output data already stored in primary memory either as "output" (e.g. to screen) or written—out to secondary memory (e.g. tape).
- 3. Perform simple arithmetic procedures (addition or subtraction) upon data in primary memory only.
- 4. Perform logical comparisons (disjunction, conjunction and negation) between two items of data in primary memory.

 (Remember I/O, arithmetic, comparisons only)

To continue the analogy with ourselves, I doubt whether we can do anything more than these operations except that we use experience. The computers' analogue of this is the PROGRAM as it possesses zero intelligence.

THE PROGRAMMING TASK is to utilize the high speed and large memory capacity of a computer system to do something useful – such as carry out calculations (number crunching), play games, monitor house security etc.

The spectrum of tasks involved in programming is very broad, so little wonder that beginners have trouble grasping the essentials, or that many programs are "badly" written. The task involves taking an idea or concept and translating that into a symbolic (program statement) form of representation. An intermediate stage in this translation often involves modelling the phenomenon being programmed. This psychologically involves moving from concrete concepts to various levels of abstraction – again a very difficult thing for, particularly young, minds to master.

The transition from an idea to a program can seldom be achieved in one leap — more often a number of intermediate steps are required. Liken it to writing an essay where drafts and notes are used before the final prose is produced. Fortunately a number of useful tools have been developed to assist in producing a good program.

In my view, one of the greatest pitfalls of the home computer boom is that these intermediate steps are not understood by Users so that, at least, bad programs and, at worst, disillusioned programmers result. Many of these people may find their way into the computer industry of the future. There is ALWAYS more personal satisfaction in achieving a "good" job even if it is only a games program for the kids. It is also more fun, (the essence of home micros) as there is less hassle in getting a program to run, and more time for more programs.

In the microcomputer environment where there are always hardware limitations, it means that it is very difficult to completely seperate hardware and software aspects of the programming task. The programmer may have to get "close to the hardware" — usually due to hardware/memory limitations or restricted I/O capabilities. Don't shy away from hardware by saying "but I am only interested in writing programs" as the two are somewhat inseperable.

Next month we will look at the various stages in the programming task, or how to approach a programming exercise. (see, no mention of BASIC code in this article!)

Finally, I would like to offer to Users that your programming queries will be

answered if you write to me - with a SAE. please. In this manner you should get what you want and I will obtain a feel for the type of problems Users in the Hunter Valley are experiencing.

Write to Bob Kitch, 7 Eurella St., KENMORE, Qld. 4069

As mentioned in Part 1 of this series, the programming task is a large and complex feat of organization and requires a wide range of skills. It is possible, and best, to break the task down into six segments - each of which must be thought about, planned and then carried out to ensure the successful completion of a software project. Even a small program requires that a cursory consideration of the six segments be made — although some of them may be quickly passed over as trivial. But it is certain that larger programs (more than 200 lines) require careful planning for success.

Before describing the six steps, it is worth thinking about "What makes a

GOOD program?"

A program may be judged from a number of different standpoints; each is not necessarily mutually exclusive and sometimes some conflicts require that a trade-off be made.

The first criteria is that a program should be EFFICIENT. Efficiency can be considered from a number of varying view points. For example, optimization of the run-time can be considered as efficient. Also, reduction in storage requirements for both program code and variables can be considered as efficient programming. Furthermore, and particularly if one is developing software commercially, then efficiency can be measured in terms of the actual time required to get an applications program running and the ease of maintenance of that code. The use of appropriate data types and data structures can greatly improve the efficiency of a program. The selection of a suitable algorithm can also assist. Finally, ease of debugging so that the program can be updated or modified may be considered desirable.

The second criteria is GENERALITY and it is here perhaps that so many programs "score" so poorly. Rather than a program being written to solve a particular chore, it should be broadly written to handle a wide range of problems. The use of subroutines and functions developed and debugged previously can enormously improve programming productivity. Often a simple sustitution of a variable for a constant in a program can broaden the the applicability of the program significantly.

The final criteria is ELEGANCE, which is a little harder to both define and achieve. An elegant program is one that is simple and ingenious, and possibly uses an algorithm or data structure that may not be immediately obvious to the application. The so-called "programmer's tricks" are often elegant solutions to a programming problem; but beware, some are attempts by programmers to conceal their programming stategy.

These then, are general guidelines to try and attain in your programming and by which to judge a particular programming effort as good, mediocre or poor. Notice that they are not language specific comments and are equally applicable to any programming language or exercise.

To return to the six steps in the programming task — I will briefly discuss each in turn and ask that you consider each one when embarking upon your next programming exercise. Also as one proceeds through the steps, it is often necessary to recycle back through some of the preceeding steps, to iteratively improve the exercise and your understanding of ideas.

1. PROJECT SELECTION. This may appear trivial, but we all have too many ideas for programs and rarely know which one to tackle next. Also be honest with yourself; some of the projects are probably too ambitious for your existing skills and an attempt upon these will possibly result in frustration and perhaps failure. Choose an exercise that is challenging and worthwhile. Try not to "reinvent the wheel", try to be aware through reading magazines or discussing with other Users what programs are already available. Modifying an existing program to suit your specifications is sometimes quicker - it also allows you to study how other programmers tackle problems. O.K., so now you have an idea or problem that you wish to tackle and solve.

2. PROJECT FEASIBILITY. Again be honest. Do you have the hardware, software and know-how to achieve the result? Its is not really much use trying to write large business-oriented data base programs for an 8K taped-based VZ! Check that

the task is reasonable.

3. PROJECT DEFINITION. This is where the idea starts to get translated into a reality. It is also the phase where generality can be written in. It is easiest to start by thinking about the input to the program. Is it keyboard oriented, or is it to come from a programmble I/O port? Perhaps the program reads only DATA statments to configure itself or maybe the program must check if a printer is connected to the sytem? Start defining what the input will look like. Assign variable names with meaningful mnemonic names at this stage also.

Next, define the output expected from the program. Is it to write to tape and in what format? Perhaps it is to be screen oriented — can sound be used — or perhaps voice synthesis to tell the operator what is going on? Plan very carefully and fully the layout of the expected output as this is how Users will

initially perceive the quality of the program.

After defining the I/O for the program we should now have a feel for the anticipated range of parameters that the program is meant to accept and also handle. This brings in the very important concept of defining the BOUNDS within which the program must function correctly. Following on from this, is range checking of all input parameters so that the program cannot go beyond the range that it was designed for and give unexpected results. A number of warning messages must be built into the program along with error capture and recovery routines. It is failure to define the operating bounds of a program that causes most crashes or rogue behaviour. Even the definition of integer variables at this stage can assist by improving program execution time and reducing storage requirements.

The definition stage should be roughed out on pieces of paper kept for later reference. Perhaps better, is to use an old exercise book. Another benefit of this is that over a period of months your progress can be measured and your growth of programming ideas recorded. Another benefit (although I hardly dare mention it!) is that if, after the coding stage, a system crash occurs and you didn't SAVE the program, then all is not lost — at least an outline of the program remains.

4. DESIGN PHASE. Having sorted out I/O and operating bounds, the actual selection of an algorithm to achieve the result is commenced. By this time some idea of the number of variables required and their type should have begun to gel. This is also the stage where your basic honesty in stages 1 and 2 may catch up with you! Data structure organisation and algorithm selection are really experience-related skills — hence the suggestion to read and/or modify existing programs. But do not despair — practice makes perfect.

5. IMPLEMENTATION PHASE. To date very little actual coding should have been done; in fact the computer need not even have been turned on! Some people may be surprised at how late in the task the computer actually enters into the picture. An awful lot of planning and organizing can be done off the computer and on the

"backs of old envelopes".

It is also at this stage that the choice of programming language should be made. Is the program time dependant? If it is, then it should probably be written is Assembler. If the actual timing is not so critical then writing in BASIC with its diagnostics and helpful features (so typical of a high level language) deem it sensible. Experienced programmers will probably use a bit of each in practice. A very sensible compromise is to develope the program in interpreted BASIC and once finalized and debugged, compile the BASIC code to speed up execution.

6. EVALUATION PHASE. This is the moment of truth! Does the program fulfil all the criteria set out in the definition phase. If so, then you have successfully achieved your task. Is the output as you expected it? Are the results correct? It is a good idea to have a standard set of data to exercise the program so that it can be quickly verified after a program alteration. Ensure that all logical paths through the program have been exercised so that no spurious errors of logic

8

remain undetected. Finally, deliberately try values that are out of the intended bounds of the program to ensure that you have trapped them and that the program recovers from this type of misuse above and beyond its' intended design range.

CONTINUED FROM PAGE 14

ľ

0561*VZ200 GRAND PRIX *	0590 DEFW 8686 H
0562M3G2 EQÚ \$	0591 DEFW 8686H
O563*BY P. HICKMAN	0592 DEFW 8606H
Opposit 1. Ministra	0593 DEFN 8686H
0564MSG3 EQU \$	0594 DEFN 8686H
0565*** FUNCTION KEYS ** *	0595 DEFN 8686H
0566MSG4 BQU \$	
0567* Q = INCREASE SPICED	0596 DEFN 6686H 0597 DEFN 8686H
O568MSG5 RQU \$	0598 DEFN 0606H
0569* A = DECREASE SPEED	0599 DEFN 8686H ·
O570MSG6 EQU \$	0600 DEFW 9696B
O571* M = MOVE LEFT *	0601MS14 EQU \$
0572MSG7 BQ8 \$	0603*YOUR SCORE '=*
Q573* , = MOVE RIGHT *	
0574MSG8 EQU \$	O603MS15 DQU \$
0575*PRESS B TO ENTER BASIC	0604*PRESS ANY KEY TO CONTINUE
0576MGG9 BQU \$	0605MS16 EQU \$
0577*PRESE ANY KEY TO START	0606*YOU ARE THE CHAMPION
057816310 190 \$	OBO7; ####################################
O579*CHAMPION SCORE = *	0608; IMPROVIDMENTS NEEDED
0580M511 EQU \$	0609; ####################################
0581*2ND REST SCORE = *	0611; ADD BREAK TO RESTART
0582MS12 EQU \$	ARIT AND BEACK TO MISTERS ART
0583*3RD BEST SCORK = *	0613; ADD MAX NO OF CARS = 5
Onco-orn Brot cooks -	0614; ADD 5 LEVELS OF PLAY
0584MS13 DEFN 8686H	0615; HIGHEST LEVEL HAB NORE
O585; VERGE DESIGN	ORIG; OBSTACLES, FASTER SPEED,
0586 DEFW 8686H	0617; AND NARROWER BOAD
0587 DEFW 8686H	0616; ADD TITLE SCREEN
O588 DEFW BEBEH	0619; DISABLE BLOAD COMMAND
0569 DEFN 8686H	Antalanda news Antaran

HISTORY

Reprinted from the MPCUG journal PC UPDATE.

Author Ian McDowell

Australia possessed only three electronic digital computers in the late 1950s;

.CSIRAC at Melbourne University

SILLAC at Sydneu University

.WREDAC at the Weapons Research Establishment at Salisbury

CSIRAC is now a musemum piece, and you may see it at Monash University Chrisholm Campus in Caulfield.

paper input.It possessed CSIRAC used punched addition, subtraction and ultiplication resident routines. other required copying library tape segments to the user's input roll. Logical circuits used thermionic valves, the transistor had line held appeared but not been mobilised. Mercury delay information. Cathode ray tubes displayed the bytes contained in fifteen storage registers. A Friden Flexowriter read and printed punched paper tape output. CSIRAC had about the power of a T159 Instruments programmable calculator. Nonetheless, on the word of command DO, it produced RESULTS. It freed us from slow and noisy electro-mechanical machines. Learned persons produced all sorts of worth-while discoveries on all three computers. first conference on automatic computing and data processing in Australia at Sydney University inn May 1960 gave oppertunity to share these discoveries. Presenters gave a total of 157 papers; 42 on commercial applications, 65 on technical applications, 41 on design and programming techniques and 9 describing other computers soon to be offered by commercial firms. This writer spoke to a technical paper on a problem solved using CSIRAC which various learned journals later published in Australia and overseas. The rise of electronic digital programmable computing seems rapid to it's youthful euthusiasts, but it has taken a third of a century to reach it's present level, andit's early proponents look for the 486 in the geriatic ward.

Postcript by Peter Smith, editor of MPCUG PC UPDATE.

I too have fond memories if CSIRAC--my introduction to computing. I recall those huge festoons of 12-track paper tape and the desire, and need, to cram more and tighter code into it's 768 words of memory(20 bits each, yes the "BYTE"-a word not yet invented-was then only 5 bits long). I compare my portable, battery powered trlrphone, weighing a few ounces, to the tons and cubic yards of CSIRAC, needing a small power station to drive it, and realise that the phone has more memory than CSIRAC and is much more user friendly! I have long since decided that only knaves and fools dare predict longer than about 5 years in this industry.

GRAND PRIX

0002: MRITTEN IN 1687 0003: PRIME J. HICKMAN 0004: PRIME J. HICKMAN 0004: PRIME J. HICKMAN 0006: PRIME J. HICKMAN 0006: PRIME J. HICKMAN 0006: PRIME J. HICKMAN 0007: A THE: -80 PUNDINAM 0010: STATE T. THE TABLE AS 0011: THE CRIGIN FOR YOUR PROGRAM 0011: THE CRIGIN FOR YOUR PROGRAM 0012: ADD EAST PARAMETER MS3 0016: SELECT MODE(O) 0014: GREEN PROGRAM 0018: SELECT MODE(O) 0014: GREEN PROGRAM 0018: SELECT MODE(O) 0019: THE TOM A COURT PROGRAM 0018: THE TABLE AS 0019: SELECT MODE(O) 0019: THE TOM A CARGINUMS COLOUR 0019: THE TOM A CARGINUMS COLOUR 0019: THE TOM A CARGINUMS COLOUR 0022: LD (7898), A 0023: LD (7898), A 0023: USR ADDRELL, SADR 0030: LD (7898), A 0023: USR ADDRELL, SADR 0030: LD (7898), A 0031: T. D. POINTER 0032: DEC HL 0033: CLEAR SO BYES 0034: CLEAR SO BYES 0035: T. D. TORAGEN, ML 0036: SCALL FRANCOLUMN 0037: LD TORAGEN, ML 0038: CLEAR SO BYES 0039: CLEAR SO BYES	CONT. VE PRIE VERSTON 1	
0004; PEM MICHO 80 MAGAZINE 0005; SASSEREE 7 (JURE 1992) 0006; SASSEREE 7 (JURE 1992) 0006; SASSEREE 8 SASSEREE 8 SASSEREE 8 0072; DEC NOT 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	O(P)Z;WEITTEN IN 1987	
0005; Vol. 3 minute BV 7 minute 1982) 0006; Wat The Seas Principle 1982) 0006; A The Seas Principle 1982 of 10072; DRAW VERGE & BOAD ON LINE 1 0000; SET YOUR OWN START ADDRESS of 10:35 THAT IT IS THE SHAME AS 0075; OLDY LM I TO ALL SCREEN 0075; DRAW VERGE & BOAD ON LINE 1 0000; SET YOUR COME START ADDRESS of 10:35 THAT IT IS THE SHAME AS 0075; DRAW VERGE & BOAD ON LINE 1 0011; THE ORIGIN FOR YOUR PROGRAM 0075; DRAW VERGE & BOAD ON LINE 1 0012; ADDRESS OF 10:0000 0075; DRAW VERGE & BOAD ON LINE 1 0013; DRAW START WH3 0075; DRAW VERGE & BOAD ON LINE 1 0015; SET SHIGHT TO OCCOCH 0075; DRAW VERGE & BOAD ON LINE 1 0015; SET SHIGHT TO OCCOCH 0075; DRAW VERGE & BOAD ON LINE 1 0015; SELECT WORKSHAME SHEET BY SHEET SHEET BY SHEET BY OVER LINE 0075; DRAW VERGE & BOAD ON LINE 1 0015; SELECT WORKSHAME SHEET SHEET SHEET BY OVER LINE 0075; DRAW VERGE & BOAD ON LINE 1 0015; SELECT WORKSHAME SHEET	0003; BY PETER J. HICKMAN	66 70:
OOOS; SREWSERSENSENSENSENSENSENSENSENSENSOON; A TREE SO PETAGRAM OOOS; SREWSERSENSENSENSENSENSENSENSENSENSENSENSOON; SREWSENSENSENSENSENSENSENSENSENSENSENSENSENS	0004; FROM MICHO 80 MAGAZINE	0071; Clear Screen
0007; A THE-SO PENGRAM 0008; ENT YOUR OWN SYART ADDRESS 0010; ENT ORIGIN TO COCOCH 0014; SET PARAMETER N=3 0015; ENEMBER ON COCOCH 0014; SET PARAMETER N=3 0016; ERLECT MODE(C) 0017ENTS KOR A 0016 LD (789M1), A 0018 CREMAN, TRINTA 0018 CREMAN, TRINTA 0019 CALL DATE ORIGINA 0022; ONDEAL DATE ORIGINA 0023; OUTPUT DEVICE = VIDEO 0024 LD (789M1), A 0025 INC ABDRESS 0026 LD (789M1), A 0027 LD (789M1), A 0028; OURPUT DEVICE = VIDEO 0029 LD (789M1), A 0021; T. O. POTTER 0029 LD (789M1), A 0021; T. O. POTTER 0031, T. O. POTTER 0031, T. O. POTTER 0032 DEC NA 0033 LD (781M1), HL 0034; CLEAR SO BYTES 0035 LD DE, OFFCRM 0036 ADD HL, DK 0037 LD (ACAD), HL 0037 LD (CARN), HL 0036; CTAL TRINTA 0036; CTAL ROWN 0037 LD (CARN), HL 0037 LD (CARN), HL 0038; CTALT CONTROL 0039 LD (CARN), HL 0039 LD (CARN), HL 0030; CAR START FORMER 0034 LD (CARN), HL 0035 LD DE, OFFCRM 0036 ADD HL, DK 0037 LD (CARN), HL 0037 LD (CARN), HL 0038; CTALT SUPENBOR 0039 LD (CARN), HL 0030 LD (CARN), HL 0039 LD (C	OCOS, ACT 2 WANTERS A (20WE 1885)	0072CLRS CALL O1COH
0000; #################################	OOO) A TREAD PROPERTIES TO SERVICE SER	0079; DRAW VERGE & BOAD ON LINE 1
0019; SET YOUR OWN START ADDRESS OOIT; THE ORIGIN FOR YOUR PROGRAM OOIT; THE ORIGIN FOR YOUR PROGRAM OOIT; DRAW CALL COPY 0012; SET DRESS OOIT; THE ORIGIN FOR YOUR PROGRAM OOIT; DRAW CALL COPY 0012; SET DRESS OOIT; SERVIN FOR O	(1000) : 其并實際或其實際有有於發展的	0074 CALL DRAW
0011;30 THAT IT IS THE SAME AS 0011;THE ORIGIN FOR YOUR PROGRAM 0012SADE EQU 0COOCH 0014;32T PARAMETER N=3 00079; SCROLL BOWN BY ONE LIME 0044;32T PARAMETER N=3 00615; CELLECY MODE(O) 00616; CELLECY MODE(O) 0062; CELLECY MODE(O) 0063; CEROLL SOWN BY ONE LIME 00615; CELLECY MODE(O) 0063; CEROLL SOWN BY ONE LIME 00615; CELLECY MODE(O) 0063; CEROLL SOWN BY ONE LIME 00615; CELLECY MODE(O) 0063; CEROLL SOWN BY ONE LIME 00615; CELLECY MODE(O) 0063; CEROLL SOWN BY ONE LIME 00615; CELLECY MODE(O) 0063; CEROLL SOWN BY ONE LIME 00615; CELL LD HI, VIDE+479 0063; LD ER, VIDE+411 0063; LD ER, VIDE+411 0063; LD ER, VIDE+411 0063; LD ER, VIDE+411 0063; CEROLL SOWN BY ONE LIME 0066; CELL RAMD 0067; SCROLL SOWN BY ONE LIME 0068; CEROLL BOWN BY ONE LIME 0068; CEROLD BOWN BY ONE LIME 0068; CEROLD BOWN BY ONE LIME 0069; CEROLL BOWN BY ONE LIME 0069; CEROLD BOWN BY ONE LIME 0069; CERO	OUUS EET YOUR OWN START ADDRESS	
0011; THE OMIGIN FOR YOUR PROGRAM 0012SADE EQU COCOOH 0013; SET ORIGIN TO OCOOH 0014; SET PARAMETER M:3 0015; ### OF PARAMETER M:3 0015; ### OF PARAMETER M:3 0015; ### OF PARAMETER M:3 0016; ### OF PARAMETER M:3 0006; ### OF PARAMETER M:3 0006; ### OF PARAMETER M:3 0006; ### OF PARAMETER M:3 0007; ### OF PARAMETER M:3 0008; ### OF PARAMETER M:3 0009; ### OF PARAMETER M:	ODIO; SO THAT IT IS THE SAME AS	
001:25ADE EQU OCOOCH 0014;SET PARAMETER N -3 0014;SET PARAMETER N -3 0015;SET PARAMETER N -3 0016;SED CRICKIN ENDN BY ONE LINE 0016;SELECY MODE(O) 0016;SED CRICKIN ENDN BY ONE LINE 0017;SELECY MODE(O) 0018;SONSAL FEIRT 0020 LD (7894H), A 0021;SONSAL FEIRT 0022 LD (7814H), A 0023;SUTFUT DEVICE = VIDEO 0024 LD (7894H), A 0023;SUTFUT DEVICE = VIDEO 0026;SONSAL INFOT 0027 LD (7894H), A 0028;USR ADDRESE 0029 LD (7896H), A 0021;SONSAL INFOT 0021 LD (7896H), A 0023;CITER ADDRESE 0024 LD (7896H), A 0030 LD (7968H), BL 0031;T.O.M POINTER 0032 LD (7968H), BL 0033;CT.C.AR SO (881H), BL 0034;CLAR SO (881H), BL 0035;CT.C.AR SO (881H), BL 0036;CT.C.AR SO (881H), BL 0036;CT.C.AR SO (881H), BL 0037 LD (7968H), BL 0038;CT.C.AR SO (881H), BL 0039;CT.C.AR SO (881H), BL 0044;CAR START FORITION 0040;CALL GOSGH 0041;CAR START FORITION 0041;CAR START FORITION 0042;CT.C.AR SO (881H) 0044;CALL GOSGH 0044;CALL GOSGH 0044;CAR SCAREN 0055;MITTAL REDUCTION 0046;LD DE, VIDE; 0047;HITTAL SCORE 0055;MITTAL REDUCTION 0056;MITTAL REDUCTION 0057;SORT SURRES 0058;MITTE RESERVANCE 0059;MITTE LINEADDICTION 0069;CALL FROM 0079;MITTE LINEADDICTION 0069 0060;CALL FROM 0071;HITTAL SCORE 0060;CALL FROM 0079;CALL GOSGH 0010;CALL GOSGH 0011;SORVERS 0110;SORVERS 0110;SORVERS 0110;SORVERS 0110;SORVERS 0110;CR 0111;CR 01	OOII; THE ORIGIN WOR WITH PRIVADAM	
0015: SELECT PARAMETER NOOD 0015: SELECT MODICAL 0016: SELECT MODICAL 0016: SELECT MODICAL 0016: SELECT MODICAL 0016: SELECT MODICAL 0017: MODICAL 0017: MODICAL 0018: MODICAL 0018: MODICAL 0018: MODICAL 0019: MODICAL 0019: MODICAL 0020: LD (78181), A 0021: MODICAL 0022: LD (78181), A 0023: OUTPUT DEVICE = VIDEO 0024: LD (78961), A 0025: INC A 0026: MODICAL 0027: LD (78961), A 0028: USR ADDEESS 0029: LD HL, SADR 0030: LD (78981), M 0031: LD (78981), M 0031: LD (78981), M 0032: DEC HL 0033: LD (78981), M 0033: LD (78981), M 0034: CLEAR SO BYTES 0036: ADD HL, DK 0037: LD (78081), HI 0038: START GAME: PADEK 0039: LD DE, OFFCEH 0040: CALL OGGGI 0041: CAR START FOSITION 0041: LD DE, VIDE+438 0041: LD DE, VIDE+438 0041: LD DE, VIDE+438 0042: LD (78081), DK 0044: LD DE, VIDE+438 0045: INTIAL SCHEEN 0046: LD DE, VIDE+438 0047: INITIAL SCHEEN 0047: INITIAL SCHEEN 0055: CLEAR SCHEEN 0056: CALL FRAM 0056: CALL FRAM 0057: SURT SCURES 0058: GALL SORT 0059: SCHER BL, CE IN 'B' PRESSED 0056: GALL SORT 0057: SURT SCURES 0058: GALL SORT 0059: SCHER BL, CE IN 'B' PRESSED 0060: CALL, FRAM 0074: CLEAR SORES TO SCHEEN 0056: GALL STAP 0066: JP NZ, CLES 0066: JP NZ, CLES 0066: JP NZ, CLES 0066: JP NZ, CLES 0067: CALL SAMP 0074: CLEAR SORES TO SCHEEN 0075: SURT SCURES 0066: JP NZ, CLES 0066: JP NZ, CLES 0067: CALL SAMP 0077: SURT SCURES 0078: CALL SAMP 0079: CA	OOISEADR EQU OCOOOH	
0015 SEEP NOT SECULD HIL, VIDE-479 0016 SELECT MODE(C) 0017 WITE KOR A 0016 SELECT MODE(C) 0017 WITE KOR A 0016 SELECT MODE(C) 0017 WITE KOR A 0018 SELECT MODE(C) 0017 WITE KOR A 0018 SELECT MODE(C) 0018 SELECT MODE(C) 0019 SELECT MODE(C) 0019 SELECT MODE(C) 0020 LD DR, VIDEO 0020 LD DR, VIDEO 0021 SOUTHUT DEVICE SELECT 0022 LD (7018H) A 0023 SUTPUT DEVICE SELECT 0024 LD (769GH), A 0025 SELECT MODE(C) 0026 SELECT REGIST 0027 LD (769GH), A 0029 LD DR, FORESTEN 0029 LD CALL SELECT 0029 LD TREE SELECT 0030 LD (769GH), M 0031 SELECT MODE(C) 0031 SELECT MODE SELECT 0032 LD CALL SELECT 0033 LD (768GH), M 0034 SELECT MODE SELECT 0035 LD DE, OFFCEN 0036 SELECT MODE SELECT 0037 LD (76AGH), HI. 0038 SELECT MODE SELECT 0039 LD TREE SELECT 0039 LD TREE SELECT 0040 CALL OASH 0040 SELECT MODE SELECT 0050 SELECT MODE SELECT 0040 CALL OASH 0041 SELECT MODE SELECT 0050 SELECT MODE SELECT 0050 SELECT MODE SELECT 0050 SELECT MODE SELECT 0051 LD DE, VIDE-486 0040 CALL OASH 0041 SELECT MODE SELECT 0050 SELECT MODE SELECT 0051 LD DE, VIDE-486 0051 LD DE, VIDE-486 0052 LD TYSCR), DE 0052 LD SELECT MODE SELECT 0055 SELECT MODE SELECT 0056 SELECT MODE SELECT 0057 SELECT MODE SELECT 0058 CALL SEREN 0059 SELECT MODE SELECT 0059 WELTE SOURCE SELECT 0059 SELECT MODE SELECT 0050 SELECT MODE SELEC	OO13; SET ORIGIN TO OCCOOCH	
O015:SELMCT MODE(C) O0174NTE KON A O021 LD (7818H) A O022 LD (7818H) A O022 LD (7818H) A O023 LD (7818H) A O023 LD (7818H) A O023 LD (7896H) A O024 LD (7896H) A O025 LD K, (FOGN) O027 LD (7896H) A O028 LD K, (FOGN) O029 LD HL, SADR O030 LD (7896H) A O030 LD (7896H) A O031 LD (7896H) A O032 DEC HL O031 LD (7896H) AH O032 DEC HL O033 LD (7896H) AH O034 CLEAR SO BYTES O035 LD DE, OFFCRH O036 ADD HL, DR O036 ADD HL, DR O036 ADD HL, DR O036 ADD HL, DR O037 LD (7800H) HL O038.START GAMR : PADEK O038.START GAMR : PADEK O039 LD BE, OFFCRH O040 CALL OOSOH O040 CALL OOSOH O041 LD DE, VIDE-438 O041 CAR START FOSTION O044 LD DE, VIDE-438 O045 LD DE, VIDE-438 O046 CALL OOSOH O047 INITIAL BECKNI O047 INITIAL BECKNI O049 LD CARL), DR O049 LD CARL), DR O049 LD CARL), DR O040 CALL DRAW O050 INITIAL BOWN O050 INITIAL BOWN O050 INITIAL BOWN O051 INITIAL BOWN O050 CALL FRAM O050 CALL FR	OULE : SEER SEERS SEE SEEN SEE SEE SEE SEE SEE SEE SEE S	
0017ENTE KOR A 0016 LD (789EH), A 0019; RORMAL FEINT 0020 LD (781HI), A 0021; HOEMAL BACKGIBOUND COLDING 0022; LD (781HI), A 0023; OUTPUT DEVICE = VIDEO 0024 LD (789EH), A 0025 INC A 0026; SINC BACKGIBOUND COLDING 0026; SINC BACKGIBOUND COLDING 0027 LD (789EH), A 0028; LEFT GRIGHT 0029 LD (789EH), A 0029 LD (789EH), BC 0030 LD (789EH), BC 0031; T.O. M. POINTER 0031; T.O. M. POINTER 0032 DEC H. 0033 LD (781HI), BC 0034; CLEAR 50 BYTER 0035 LD DE, OFFCEH 0036; START GANRY PADEE 0036; TABT GANRY PADEE 0037 LD (780HI), BC 0038; LEFT GRIGHT 0039 LD (780HI), BC 0030 LD (780HI), BC 0031 LD (780HI), BC 0032 LD (780HI), BC 0033 LD (780HI), BC 0034; CLEAR 50 BYTER 0035 LD DE, OFFCEH 0036; TABT GANRY PADEE 0036; TABT GANRY PADEE 0036; TABT GANRY PADEE 0040; CALL OGOGH 0041; CAR START POSITION 0041; CAR START POSITION 0041; LD DE, VIDE+496 0044; EADA START POSITION 0045 LD DE, VIDE+496 0044; EADA START POSITION 0046 LD DE, VIDE+496 0047; INITIAL SPEED 0055; LD (VYCR), DE 0056; WRITE INTRODUCTION 0056; WRITE INTRODUCTION 0057; RANDEN ER BLATT 0060 ALL CARL) 0050 CALL, ZAND 0050; CLEAR SCHEEN 0050; WRITE INTRODUCTION 0050; WRITE RECORDES TO SCENEEN 0050; WRITE WRITE BOOKES TO SCENEEN 0050; WRITE BOOKES TO SCENEEN 0050; WRITE WRITE BOOKES TO SCENEEN 0050; WRITE BOOKES TO SCENEEN 0050	OO16:CECMOW MONVON	OO62 LD DE, VIDE+511
0016 LD (789H), A 0085 HALT 0020 LD (781H), A 0085 HALT 0021; NORMAL FRINT 0022 LD (781H), A 0087; RANDELY BEND ROAD TO 0021; NORMAL BARCHBOUND COLOUR 0022; LD (789CH), A 0090 CALL RAND 0024 LD (789CH), A 0090 CALL RAND 0025 LD DR (789CH), A 0093 CP 01 0026; SUR ADDEES 0027 LD (789CH), A 0093 CP 01 0028; USR ADDEES 0030 LD (789CH), A 0093 CP 01 0030 LD (789CH), B 0094 JR 2, LFT1 0030 LD (789CH), B 0095 JR PENT 0031; T.O. M. POINTER 0032 DEC HL 0033 LD (788CH), HL 0099PENT LD HL, VIDR+15 0034 LD (788CH), HL 0099PENT LD HL, VIDR+15 0035 LD DE, OFFCER 0036 ADD HL, DR 0037 LD (78AOH), HL 0099PENT LD HL, VIDR+15 0036 ADD HL, DR 0038; START GAMR PADSE 0039 LD (78AOH), HL 0103; A=F IF HL, DR 0039 LD (78AOH), HL 0103; A=F IF HL, DR 0039 LD (CALL OASOH DE) 0044 CALL OASOH DE, VIDR+488 00447; INITIAL SPEZU 0044; EOAD START FOSITION 0045 LD DE, VIDE+488 00469 LD DE, VIDE+498 00469 LD DE, VIDE+498 00469 LD DE, VIDE+498 00469 LD DE, VIDE+7 0047; INITIAL SCENE 0051 LD DE, OSPOCH 0055; SORT SCURES 0056; WRITE BEND ROAD TO LINE 1 0067; ONTO CALL DAN 0057; SORT SCURES 0058; WRITE BEND ROAD TO LINE 1 0106 CALL DAN 0110 CP OFFH 0111; IN O CHANCE OF 0116; DEAN VERGE & ROAD TO LINE 1 0121 CP 01 0122 CP 01 0123 LD (SPED), DR 0124; INCERASE SPEED 0059; WRITE BOORES TO SCEEEN 0059; WRITE BEND ROAD INPUT 0066; MRITE DOR CALL OASOH DISCURDED TO CALL RAND 0059; ONTO CALL OASOH DISCURDED TO CALL RAND 0120 CALL RAND 0121 CP 01 0121 LD 0.0000 LD HL, 10 0122 CP 01 0123 CALL ZORD 0124; INCERASE SPEED 0125 INCERASE SCOEE 0126 DEC DR 0127 DEC DR 0128; INCERASE SPEED 0129; INCERASE SPEED 0129; INCERASE SPEED 0120; INCERASE SPEED 0120; INCERASE SPEED 0121; INC DR 0122; INCERASE SPEED 0123; INCERASE SPEED 0124; INCERASE SPEED 0125 INCERASE SPEED 0126; INCERASE SPEED 0127; INCERASE SPEED 0128; INCERASE SPEED 0129; INCERAS	001/ENTR XOR A	
0029 LD (7818H) A 0021; HORMAL BACKGROUND COLOUR 0022; LD (7818H) A 0023; GUTFUT DEVICE = VIDED 0024 LD (789CH) A 0025 INC A 0025 INC A 0026; NORMAL INFUT 0027 LD (789GH) A 0028; US ADDREES 0029 LD (789GH) A 0029; LD DR. (FOSN) 0027 LD (789GH) BC 0029 LD DR. (FOSN) 0029 LD DR. (FOSN) 0021; TO. M. POINTER 0030 LD (796EH) HL 0031; T.O. M. POINTER 0032 LD (789GH) HL 0034; CLEAR 60 BYTES 0039 LD (780H) HL 0036; START GANK : PADSE 0036 ADD HL, DR 0036; START GANK : PADSE 0036; START GANK : PADSE 0040 CALL 0060H 0041; CAR START FOSITION 0041; CAR START FOSITION 0042 LD DE, VIDE-436 0044; BOAD START FOSITION 0046 LD DE, VIDE-436 0047; INITIAL BEELD 0048 LD (FOSN), DE 0049 LD (SPELD) DR 0049 LD (SPELD) DR 0040; CARLI, ORD 0055; CALL COME 0056; CALL ACOME 0057; CALR SCORE 0058; CALL OCOM 0059; CA	O()16 LD (789bH) A	
0021 NOSBAL BRYGROUND COLORD 0022 LD	0019; NORMAL PRINT	
0022 (OUTPUT DEVICE = VIDEO) 0023 (OUTPUT DEVICE = VIDEO) 0024 LD (789CH), A 0025 (NORMAL IMPOY 0027 LD (789CH), A 0028 (USR ADDREESS 0029 LD HL, SADR 0030 LD (786CH), HL 0031; T.O.M. POINTER 0031; LD (786CH), HL 0033 LD (786H), HL 0035 LD DE, OFFCH 0036 (START SAMR : PADSE 0036 START GAMR : PADSE 0036 START GAMR : PADSE 0036 START FOSITION 0041; CAR START POSITION 0042 LD DE, VIDE+7 0044 LD DE, VIDE+7 0045 LD DE, VIDE+7 0046 LD DE, VIDE+7 0046 LD DE, VIDE+7 0047; INITIAL SCORE 0049 CALL AND 0090 CALL RAND 0090 CALL RAND 0091 LD DE, (FTGM) 0093 CP 01 0093 CP 01 0093 CP 01 0094 JR 2, LFT1 0095 JR 2, LFT1 0096 JR 2, LFT1 0096 CALL OA39H 0101; A=0 IV HL-ZDE 0096 CALL OA39H 0102; A=1 IF HL-DE 0103; A=FF IV EL(OR, 0103; A=FF IV EL(OR, 0104; A=0 IV HL-ZDE 0105 CP 01 0106 GALL OA39H 0106 GALL OA39H 0107 DEC DE 0108 CALL OA39H 0109 CALL OA39H 0100 CP OFFN 0110 CP OFFN 0111 JR Z, OKOK 0111 JR Z, OKOK 0112 INC DE 0122 CP 01 023 CALL Z, ORST 0123 CALL Z, ORST 0124; INCERASE SPEED 0050 CALL AND 0060 CALL OA39H 0101; A=0 IV HL-Z 007/LFT1 DEC DE 0109/CALL RAND 0097/LFT1 DEC DE 0109/CALL OA39H 0101; A=0 IV HL-Z 0007/LFT1 DEC DE 0100/CALL AND 0100/CALL OA39H 0101; A=0 IV HL-Z 0007/LFT1 DEC DE 0100/CALL AND 0100/CALL OA39H 0101; A=0 IV HL-Z 0007/LFT1 DEC DE 0100/CALL AND 0100/CALL OA39H 0102; A=FF IV EL(OR, 0100/CALL OA39H 0100 CP OFFN 0111 JR Z, OKOK 0111 JR Z, OKOK 0112 INC DE 0100 CP OFFN 0100 C		
0023; OUTPUT DEVICE = VIDEO 0024 LD (78BCH), A 0026; NORAL, IMPOF 0027 LD (789SH), A 0028; USR ADDRESS 0029 LD HL, SADR 0030 LD (768RH), HL 0031; T.O.M. POINTER 0032 DEC HL 0032 DEC HL 0033 LD (78B1B), HL 0034; CLEAR 50 BYTES 0035 LD PL, OFFCH 0036 ADD HL, DK 0036 ADD HL, DK 0038; START GAMR: PAUSE 0036 ADD HL, DK 0037; TLD (78ACH), HL 0038; START GAMR: PAUSE 0040 CALL OBOR 0041; CAR START POSITION 0041; CAR START POSITION 0041; CAR START POSITION 0041; DD P, VIDE+488 0044; EDD DE, VIDE+7 0046 LD DE, VIDE+7 0046 LD DE, VIDE+7 0046 LD DE, NOBOR 0047; INITIAL SPEED 0040; INITIAL SPEED 0053; CLEAR SCREEN 0040 0049 LD (SPED), DR 0046; ID (FOSN), DE 0057; CULFAR SCREEN 0058 CALL FRAN 0059; CLEAR SCREEN 0059 CALL FRAN 0059; CLEAR SCREEN 0059 CALL FRAN 0059; CLEAR SCREEN 0059 CALL FRAN 0060; CLEAR SCREEN 0059 CALL FRAN 0120 CALL DRAN 0116; OBSACLE OR BOAD 0057; CULFAR SCREEN 0059; CLEAR SCREEN 0059 CALL FRAN 0120 CALL PAND 0116; OBSACLE OR BOAD 0059; CLEAR SCREEN 0060 CALL FRAN 0121 LD DR, (SCREEN 0130 LD HL, (VICCR) 0131 LD DR, (SCREEN 0131 LD DR, (SCREEN 0130 LD HL, (VICCR) 0133; REYFORDE SCANNING EDUTINE	OOST; HOMMAL BACKGROUND COLOUR	OORS: LEFT OR RIGIST
0024 LD (75BCH), A 0090 CALL RAND 0092 LD 075BCH), A 0092 LD DK, (FOSM) 0027 LD (75BGH), A 0092 LD DK, (FOSM) 0027 LD (75BGH), A 0092 LD DK, (FOSM) 0028 (USR ADDREESS 0029 LD HL, SADR 0093 LD (76BGH), HL 0093 LD (76BGH), HL 00931; T.O.M. POINTER 0094 JR 2, LFT1 00931; T.O.M. POINTER 0095 LD DE, OFFCEH 0093 LD (76BGH), HL 00931; T.O.M. POINTER 0094 JR 2, LFT1 0097; KEKF ROAD ON SCREEN 0094 JR 2, LFT1 0097; KEKF ROAD ON SCREEN 0095 LD DE, OFFCEH 0101; A=0 IF HL-DE 0096; KEKF ROAD ON SCREEN 0101; A=0 IF HL-DE 0096; TI DE C DE 0105 CP 0100; COMPARE HL WITH DE 0098; KEKF ROAD ON SCREEN 0101; A=0 IF HL-DE 0102; A=1 IF HL-DE 0102; A=1 IF HL-DE 0103; A=FF IF HL (DK 0036; START GAME; PANDSE 0104 CALL 0A36H 0105; A=FF IF HL (DK 0036; START ROSITION 0106 JR 2, LEST 0	OU23 OUTSILE DESIGN - MARCO	
OU25 INC A 100 C) OU25 (NO MAIL INPUT O027 LD DR, (POSN) O027 LD (7896H), A 0093 CF 01 O027 LD MR, (POSN) O028 (US.R ADDEESS O029 LD ML, SADR O030 LD (796RH), ML O030 LD (796RH), ML O031; T. O. M. POINTER O032 DEC HL O033 LD (788HH), ML O031; T. O. M. POINTER O032 DEC HL O033 LD (788HH), ML O034; CLEAR 50 BYTES O100; CMPARE BL WITH DE O034; CLEAR 50 BYTES O100; COMPARE BL WITH DE O036 ADD HL DE OFFCRH O101; A= D ML DE O036 ADD HL DE O036; START GAMK : PADSE O103; A= T ML LDE O102; A= T ML LDE O103; A= T ML LDE O103; A= T ML LDE O104 CALL OA39H O104; CAR START POSITION O104 CALL OA39H O104 (CARL), DE O105; A= T ML LDE O106 JR Z, LFST O107 DEC DE O106 JR Z, LFST O107 DEC DE O106 JR Z, LFST O107 DEC DE O106 JR Z, LFST O106 JR Z, LFST O107 DEC DE O106 JR Z, LFST O107 DEC DE O107 DEC DE O107 DEC DE O108 JR Z, LFST O108 JR Z, LFST O109 JR Z, LFST O116 JR Z, DEST D116 JR Z	0024 In tracts	the state of the s
0027 LD (7896H), A 0093 CF 01 0028 (USR ADDRESS 0094 JR Z, LFT1 0029 LD RL, SADR 0095 JR DE 0096 JP 00696	Q025 INC A	0091 LD A, E
0027 LD (7699H), A 0093 CP 01 0028; USR ADDRESS 094 JR Z, LFT1 0029; LD (768FR), ML 0095 INC DE 0966 JR PENT 0030 DEC HL 0093; LD (768BH), HL 0004; LET DE OE 098; KEKP ROAD ON SCREEN 0031; T.O.M. POINTER 0007/LFT1 DEC DE 098; KEKP ROAD ON SCREEN 0032 DEC HL 0009; LD (768BH), HL 0009; LET DE OE 0FFCRH 0101; A=0 DF HL-DE 0003; LD (76A0H), HL 0102; A=1 IF HL-DE 0102; A=1 IF HL-DE 0103; A=FF IF HL-OR 0104; CALL 0A39H 0105 CP 01 0064 CALL 0A39H 0106 CALL 0A39H 0106 CALL 0A39H 0107 DEC DE 0108 CALL 0A39H 0109 CALL 0A39H 01109 CALL 0A39H 0109 CALL 0A39H 0109 CALL 0A39H 0109 CALL 0A39H 01109 CALL 0A39H	OOZ6; NORMAI, INPUT	OO9E LD DR. (POSN)
0030 LD (786RH), HL 0031; T.O. M. POINTER 0032 DEC HL 0033 LD (78BH), HL 0034; CLEAR 50 BYTES 0035; LD DE, OFFCEH 0036 ADD HL, DE 0037 LD (78AGH), HL 0038; START GAMR: PADSE 0040 CALL OBSOH 0041; CAR START POSITION 0042 LD DE, VIPE+498 0043; LD DE, VIPE+498 0044; BOAD START POSITION 0045 LD DE, VIPE+7 0046 LD DE, VIPE+7 0047; INITIAL SPEED 0047; INITIAL SCORE 0055; INITIAL SCORE 0055; INITIAL SCORE 0055; WRITE INTENDUCTION 0056; WRITE INTENDUCTION 0056; WRITE SCORES 0059; WRITE GOORES TO SCREEN 0056; PAD, CALL SGRT 0056; PAD, CALL SGRT 0056; PAD, CALL SGRT 0056; PAD, CALL SGRT 0057; SORT SCORES 0058; CALL SGRT 0059; WRITE GOORES TO SCREEN 0050; CALL, FTA 0060; PAD, CALL SGRT 0056; PAD, CALL SGRT 0057; SORT SCORES 0058; CALL SGRT 0059; WRITE SCORES TO SCREEN 0050; CALL, FTA 0060; PAD, CALL SGRT 0050; CALL, FTA 0060; PAD, CALL SGRT 0060; CALL, FTA 0060; PAD, CALL SGRT 0060; CALL, FTA 0060; PAD, CALL SGRT 0120; NCCREASE SPEED 0121; NCCREASE SCORE 0120; NCCREASE SCORE 0120; NCCREASE SCORE 0121; NCCREASE SCORE 0122; NCCREASE SCORE 0123; NCCREASE SCORE 0124; NCCREASE SCORE 0125; NCCREASE SCORE 0126; NCCREASE SCORE 0127; NCCREASE SCORE 0128; NCCREASE SCORE 0129; NCCREASE SCORE 0124; SCAR SCORE 0124; SCAR SCORE 0124; SCAR SCORE 0125; NCCREASE 0124; NCCREASE SCORE 0126; NCCREASE SCORE 0127; NCCREASE SCORE 0128; NCCREASE SCORE 0129; NCCREASE SCORE 0129; NCCREASE SCORE 0129; NCCREASE SCORE 0129; NCCREASE SCORE 0124; NCCREASE SCORE 0124; NCCREASE SCORE 0124; NCCREASE SCORE 0124; NCCREASE SCOR	0027 LD (7896H).A	0093 CF 01
0030 LD (786RH), HL 0031; T.O. M. POINTER 0032 DEC HL 0033 LD (78BH), HL 0034; CLEAR 50 BYTES 0035; LD DE, OFFCEH 0036 ADD HL, DE 0037 LD (78AGH), HL 0038; START GAMR: PADSE 0040 CALL OBSOH 0041; CAR START POSITION 0042 LD DE, VIPE+498 0043; LD DE, VIPE+498 0044; BOAD START POSITION 0045 LD DE, VIPE+7 0046 LD DE, VIPE+7 0047; INITIAL SPEED 0047; INITIAL SCORE 0055; INITIAL SCORE 0055; INITIAL SCORE 0055; WRITE INTENDUCTION 0056; WRITE INTENDUCTION 0056; WRITE SCORES 0059; WRITE GOORES TO SCREEN 0056; PAD, CALL SGRT 0056; PAD, CALL SGRT 0056; PAD, CALL SGRT 0056; PAD, CALL SGRT 0057; SORT SCORES 0058; CALL SGRT 0059; WRITE GOORES TO SCREEN 0050; CALL, FTA 0060; PAD, CALL SGRT 0056; PAD, CALL SGRT 0057; SORT SCORES 0058; CALL SGRT 0059; WRITE SCORES TO SCREEN 0050; CALL, FTA 0060; PAD, CALL SGRT 0050; CALL, FTA 0060; PAD, CALL SGRT 0060; CALL, FTA 0060; PAD, CALL SGRT 0060; CALL, FTA 0060; PAD, CALL SGRT 0120; NCCREASE SPEED 0121; NCCREASE SCORE 0120; NCCREASE SCORE 0120; NCCREASE SCORE 0121; NCCREASE SCORE 0122; NCCREASE SCORE 0123; NCCREASE SCORE 0124; NCCREASE SCORE 0125; NCCREASE SCORE 0126; NCCREASE SCORE 0127; NCCREASE SCORE 0128; NCCREASE SCORE 0129; NCCREASE SCORE 0124; SCAR SCORE 0124; SCAR SCORE 0124; SCAR SCORE 0125; NCCREASE 0124; NCCREASE SCORE 0126; NCCREASE SCORE 0127; NCCREASE SCORE 0128; NCCREASE SCORE 0129; NCCREASE SCORE 0129; NCCREASE SCORE 0129; NCCREASE SCORE 0129; NCCREASE SCORE 0124; NCCREASE SCORE 0124; NCCREASE SCORE 0124; NCCREASE SCORE 0124; NCCREASE SCOR	VV20, VSR ADDRESS	0094 JR Z,LFT1
0031; T.O.M. POINTER 0032 DEC HL 0033 LD (78B1H), HL 0034; CLEAR 80 BYTER 0035 LD DE, OFFCEH 0036 ADD HL, DR 0037 LD (78ADR), HL 0038; START GAMR: PADSE 0058; TERT HOSITION 0041; CAR START POSITION 0042 LD DE, VIDE+436 0044; EDAD START POSITION 0045 LD DE, VIDE+436 0046 LD DE, VIDE+7 0046 LD DE, VIDE+7 0047; INITIAL SPEED 0048 LD DE, 1800H 0049 LD (SPED), DR 0050; HITTIAL SCORE 0053; CLEAR SCHEEN 0053; CLEAR SCHEEN 0054; CALL OCCH 0055; WRITE INTERDUCTION 0055; WRITE INTERDUCTION 0056; CALL SCRE 0056; CALL SCRE 0058; CALL SCRE 0059; WRITE SCORES TO SCREEN 0060 CALL, FRAM 0054 CP 66 0056; DP NZ, CLES 0066 JP 0069H 0067; NSTER B(C IF 'B' PRESSED) 0067; CALF SCREEN 0066 JP 0069H 0067; KEYPOARD ON SCREEN 0059; WRITER B(C IF 'B' PRESSED) 0066; JOSE CALF SCREEN 0067; CALF SCREEN 0067; CALF SCREEN 0068; WRITER B(C IF 'B' PRESSED) 0069; WRITER B(C IF 'B' PRESSED) 0131; INC HL 0132; GARB FOR ', 'KEY 0133; GRAPPOARD SCAMNING EQUTINE 0134; GRAPPOARD SCAMNING EQUTINE 0134; GRAPPOARD SCAMNING EQUTINE	0029 LD HL, SADR	
0032 DEC H. SOLD (78B1B), H. O099PENT LD HL, VIDE+15 O0933 LD (78B1B), H. O099PENT LD HL, VIDE+15 O100; COMFARE HL WITH DE O1036 ADD HL.DK O102; A=1 IF HL.DE O1036 ADD HL.DK O1037 LD (78A0H), HI. O103; A=1 IF HL.DE O1038; START GAMK : PADSE O105; A=1 IF HL.DE	0030 LD (798EH), HL	
0033 LD (78B1B), HL 0034; CLEAR 50 BYTES 0035 LD DE, OFFCRM 0036 ADD HL, DE 0037 LD (78A0H), HL 0038; START GANR: PADSE 0038; START GANR: PADSE 0039; START GANR: PADSE 0040 CALL OGGOH 0041; CAR START PCSITION 0042 LD DE, VIDE+436 0043 LD (CARL), DE 0044; BOAD START POSITION 0045 LD DE, VIDE+7 0046 LD DE, VIDE+7 0046 LD DE, VIDE+7 0047; INITIAL SPEED 0048 LD DE, SBOOH 0049 LD (SPED), DR 0049 LD (SPED), DR 0053; CLEAR SCHEEN 0053; CLEAR SCHEEN 0055; WRITE INTERDEDICTION 0056; WRITE INTERDEDICTION 0057; SORT SCORES 0058; WRITE SCORES TO SCREEN 0050; INITIAL SORE 0051 LD DE, VIDE 0052 CALL FRAN 0055; WRITE SCORES TO SCREEN 0056; WRITE RECORES TO SCREEN 0057; SORT SCORES 0058 JP NZ, CLES 0056 JP NZ, CLES 0057; SORT SCORES 0058 JP NZ, CLES 0058; WRITE RE(C IN 'B' PRESSED) 0059; START SCAR SCAN ING BOUTINE 0056; DOSS; CALL SCAN ING BOUTINE 0057; SORT SCORES 0058 JP NZ, CLES 0059; START FOR SCANNING BOUTINE 0056; JP NZ, CLES 0057; SORT SCORES 0058 JP NZ, CLES 0059; START SCAN ING BOUTINE 0050; START SCAN ING BOUTINE	AAAA — ———	
0034; CLEAR 50 BYTES 0035 LD DE, OFFCEN 0036 ADD HL, DK 0037 LD (76A0H), HI, 0038; START GANK: PADSK 0039; TED (36A0H), HI, 0039; TED (36A0H), HI, 0039; TED (36A0H), HI, 0039; TET LD BC, SFFTH 0105 CP 01 0040; CALL 0060H 0105 CP 01 0041; CAR START POSITION 0107 DEC DE 0042 LD DE, VIDE+496 0109 CALL 0A39H 0044; EDAD START FOSITION 0110 CP OFFM 0044; EDAD START FOSITION 0110 CP OFFM 0045 LD DE, VIDE+7 0111 JE Z, OKOK 0046 LD DE, VIDE+7 0111 JE Z, OKOK 0047; INITIAL SCORE 0048 LD DE, 1800H 01140KOK LD CPOSN), DR 0048 LD DE, 1800H 01140KOK LD CPOSN), DR 0050; INITIAL SCORE 0051 LD DE, 0000 01140KOK LD CPOSN), DR 0055; CLEAR SCHEEN 0119 LD BL, 10 0056 CALL FRAN 0120 CP 01 0057; SURT SCURES 0122 CP 01 0058 CALL FRAN 0121 CP 01 0059 CALL SGRT 0121 LD A, E 0059; WRITE SOURES TO SCREEN 0122 CP 01 0059 WRITE SCORES TO SCREEN 0124; INCERESS SPEED 01093; ENTER B1C IN 'B' PRESSED 0133; EXTERORS SCORE 0133; EXTERORD SCANNING ROUTINE 0134; SCANFING ROUTINE 0135;		
0035 LD DE, OFFCEH 0037 LD (78ADH), HL 0038; START GAMR : PADSE 0039; START GAMR : PADSE 0040 (CALL 0080H 0041; CAR START FCSITION 0042 LD DE, VIDE+498 0044; EOAD START POSITION 0045 LD DE, VIDE+498 0046 LD DE, VIDE+7 0047; INITIAL SPEED 0050; INITIAL SCORE 0051 LD DE, 1800H 0050; LD (YSCR), DE 0053; CLEAR SCHEEN 0056; WRITE INTRODUCTION 0056; WRITE INTRODUCTION 0058; WRITE INTRODUCTION 0059; WRITE SCORES 0059; WRITE SCORES 0051 CALL, FRAM 0056; WRITE SCORES 0051 CALL, FRAM 0056; WRITE SCORES 0052 CALL, FRAM 0056; WRITE SCORES 0058 CALL, SCRT 0059; WRITE GOORS TO SCREEN 0059; WRITE GOORS TO SCREEN 0050; CALL, FRAM 0120 CALL ARR 0121 CP 0160 JR Z, LES 0100 CALL ASSH 0100 CALL ASSH 0100 CALL DRAW 0111 JR Z, OKOK 0112 INC DE 0113; SAVE ROAD FOSITION 0140ROK LD (FOSN), DR 0140ROK LD (FOSN), DR 0116; DRAW VERGE & ROAD TO LIME 1 0150 CALL DRAW 0117; I IN 10 CHANCE OF 0118; SAVE ROAD TO LIME 1 0120 CALL BRAW 0119 LD HL, 10 0120 CALL BRAW 0121 CP 014 CALL OASSH 0150 CP 0160 JR Z, LES 0160 JR Z, L	1.00.207, 202.	0100; COMPARE HL WITH DE
0036 ADB HL, DE	0035 LD DE. OFFCEH	0101; A=O IV HL=DE
0038; START GAMR: PADSE 0038; START GAMR: PADSE 0030; START GAMR: PADSE 0040; CALL OOSOH 0041; CAR START POSITION 0041; CAR START POSITION 0042; LD DE, VIDE+436 0043; LD (CARL), DE 0044; ROAD START POSITION 0046; LD DE, VIDE+7 0046; LD DE, VIDE+7 0047; INITIAL SPEED 0048; LD DE, 1800H 0049; LD (SPED), DR 0049; LD (SPED), DR 0050; INITIAL SCORE 0051; LD DE, 0000 0052; LD (YSCR), DE 0054; CALL OCCH 0055; CIEAR SCREEN 0056; WRITE INTRODUCTION 0056; WRITE INTRODUCTION 0056; WRITE INTRODUCTION 0058; CALL SORT 0059; WRITE SCORES 0059; WRITE SCORES 0050; CALL, FRAN 0050 0051; ENTER B IC IN B PRESSED 0061; MAIT FOR GBOARD INPUT 0066; JP NZ, CLES 0061; CALL SCORE 0063; ENTER B IC IN B PRESSED 0066; JP NZ, CLES 0067; OOSS; ENTER B IC IN B PRESSED 0067; SORT SCORES 0066; JP NZ, CLES 0067; OOSS; ENTER B IC IN B PRESSED 0067; SORT SCORES 0066; JP NZ, CLES 0067; OOSS; ENTER B IC IN B PRESSED 0067; SORT SCORES 0068; JP NZ, CLES 0069; CALL SCORE 0060; CALL SCORE	0036 ADB HI DR	
0038; START GAME : PADSE	0037 · LD (70A0H), HL	
0040 CALL 0060H 0107 DEC DEC 0041; CAR START POSITION 0107 DEC DEC 0043 1D (CARL), DE 0109 CALL 0A39H 0109 CALL 0A39H 0044; EDAD START POSITION 0110 CP OFFH 0048 LD DE, VIDE+7 0111 JE Z, OKOK 0047; INITIAL SPEED 0108; INC DEC 0113; SAVE ROAD POSITION 0048 LD DE, 1800H 0116; DRAW VENGE & ROAD TO LINE 1 0056; INITIAL SCOKE 0116; DRAW VENGE & ROAD TO LINE 1 0056; INITIAL SCOKE 0117; I IN 10 CHARGE OF 016; DRAW 0055; WRITE INTEDUCTION 0195; WRITE INTEDUCTION 0056 CALL FRAM 0122 CP 01 0057; SORT SCORES TO SCREEN 0124; INCERASE SPEED 0059; WRITE BOORES TO SCREEN 0125 CALL Z, OBST 0059; WRITE BOORES TO SCREEN 0126 DEC DE 0063; ENTER B 1C IN 'B' PRESSED 0066 JP NZ, CLES 0131 INC BL. (YECR), DE 0132; LD (YECR), DE 0065; JP NZ, CLES 0131 INC BL. (YECR), DE 0132; INCERASE SCOKE 0130; INCERASE SPEED 0131 INC BL. (YECR), DE 0132; INCERASE SCOKE 0131 INCERASE SCOKE 0132; INCERASE SCOKE 0133; REYBOARD SCANNING BOUTINE 0134; BEFAN FOR ', ' KEY 0134; BEFAN FOR ', ' KEY 0135; CAN LD A (REFWH) A A (REFWH)	0038; START GAMR : PAUSE	
0041; CAR START POSITION 0042 LD DE, VIDE+486 0043; LD (CARL), DE 0044; BOAD START POSITION 0045 LD DE, VIDE+7 0046 LD DE, VIDE+7 0047; INITIAL SPEED 0047; INITIAL SPEED 0049 LD (SPED), DR 0049 LD DE, 1800H 0050; INITIAL SCORE 0051 LD DE, 0000 0052; CLEAR SCREEN 0054 CALL FRAN 0056; WRITE INTRODUCTION 0056 CALL FRAN 0057; CORT SCURES 0059; MRITE GOORES TO SCREEN 0050; MRITE GOORES TO SCREEN 0050; SNTKR B (C IN 'B' PRESSED) 0066 JP NZ, CLES 0067; 0068; 0068; 0068; 0068; 0068; 0068; 0068; 0068; 0069; 00		
0042 LD DE, VIDE+498 0043 LD (CARL), DE 0044; ROAD START POSITION 0046 LD DE, VIDE+? 0048 LD (POSN), DE 0047; INITIAL SPEED 0049 LD (SPED), DR 0049 LD (SPED), DR 0050; INITIAL SCORE 0051 LD DE, 0000 0053; CLEAR SCREEN 0054 CALL, O1CDH 0055; WRITE INTHODUCTION 0056 CALL FRAN 0057; SORT SCURES 0061; MAIT FOR (BOARD INPUT) 0062 CAL' SYB 0065 JP NZ, CLES 0066 JP NZ, CLES 0067; 0068; 0067; 0068; 0069; 00		0107 DEC DE
OO43 I.D (CARL), DE		0108LEST LD RL, VIDE+1
0044; BOAD START POSITION 0045 LD DE, VIDE+7 0046 LD (POSN), DE 0047; INITIAL SPEED 0048 LD DE, 1800H 0049 LD DE, 1800H 0050; INITIAL SCORE 0051 LD DE, 0000 0052 LD (YSCR), DE 0053; CLEAR SCREEN 0056 CALL FRAN 0056 CALL FRAN 0056 CALL SCORT 0059; WRITE INTRODUCTION 0059; WRITE SCORES 0060; WRITE SCORES 0061; MAIT FOR GROAD INPUT 0064 CP 66 0065; DP NZ, CLES 0067; 0066; JP NZ, CLES 0067; 0066; JP NZ, CLES 0067; 0068; CALL SCAR SCREEN 0120 CALL SCAR 0121 LD DE, 0000 0126 CALL FRAN 0122 CP 0150 CRIL FRAN 0122 CP 0160 DEC DE 0170 DEC DE 0180 DE 0180 DEC DE 0180 DE 018	,,	OLO9 CALL DA39H
0045 LD DE, VIDE+? 0048 LD (PCSN), DE 0047; INITIAL, SPEED 0048 LD DE, 1800H 0049 LD (SPED), DR 0050; INITIAL SCORE 0051 LD DE, 0000 0052 LD (YSCR), DE 0053; CLEAR SCREEN 0054 CALL, OICDH 0055; WRITE INTRODUCTION 0056 CALL FRAM 0056 CALL FRAM 0057; SORT SCURES 0059; WRITE SCORES TO SCREEN 0060 CALL, SORT 0060 CALL, STR 0061; WAIT FOR (BOARD INPUT) 0064 CP 66 0065; JP NZ, CLES 0067; 0067; 0067; 0068; 0067; 0068; 0069; 0069; 0069; 0069; 0060;	0044: BOAD START POSITION	
0047; INITIAL SPEED 0113; SAVE ROAD FOSITION 0048 LD DE, 1800H 01140KOK LD (POSN), DR 0049 LD (SPED), DR 0116; DRAW VERGE & ROAD TO LIME 1 0050; INITIAL SCORE 0116 CALL DRAW 0051 LD DE, 0000 0117; 1 IN 10 CHANCE OF 0052 LD (YSCR), DE 0116; OBSTACLE ON BOAD 0053; CLEAR SCREEN 0119 LD HL, 10 0054 CALL FRAM 0120 CALL RAND 0055; WRITE INTRODUCTION 0121 LD A, E 0056 CALL FRAM 0122 CP 01 0059; WRITE SCORES 0124; INCREASE SPEED 0124; INCREASE SPEED 0059; WRITE GOORES TO SCREEN 0125 LD DE, (SPED) 0050 CALL SORT 0125 LD DE, (SPED) 0050; WAIT FOR GOORD INPUT 0127 DEC DE 0063; ENTER B(C IN 'B' PRESSED 0126; INCREASE SCORE 0130 LD HL, (YSCR) 0056; JP NZ, CLES 0131 INC HL 0057; 0066 JP 0069H 0133; REYBOARD SCANNING ROUTINE 0134; SCAR FOR ', 'KEY 0135; CALL D A, CREEN 0134; SCAR FOR ', 'KEY 0135; CALL D A, CREEN CONSTRUCTION 0127 DEC DE 0129; INCREASE SCORE ', 'KEY 0133; REYBOARD SCANNING ROUTINE 0134; SCAR FOR ', 'KEY 0135; CALL D A, CREEN CONSTRUCTION O134; SCAR FOR ', 'KEY 0135; CALL D A, CREEN CONSTRUCTION O135; CALL D A, CREEN CONSTRUCTION O134; SCAR FOR ', 'KEY 0135; CALL D A, CREEN CONSTRUCTION O135; CALL D A, CREEN CONSTRUCTION O135; CALL D A, CREEN CONSTRUCTION O136; CALL D A, E CAL	0045 LD DE, VIDE+7	
0048 LD DE, 1800H 0049 LD (SPED), DR 005G; INITIAL SCORE 0051 LD DE, 0000 0052 LD (YSCR), DE 0053: CLEAR SCREEN 0054 CALL OICDH 0055; WRITE INTRODUCTION 0056 CALL FRAM 0056 CALL FRAM 0057; SORT SCORES 0059; WRITE GOORES TO SCREEN 0059; WRITE GOORES TO SCREEN 0060; INAIT FOR BOARD INPUT 0063; ENTER B (C IF 'B' PRESSED) 0065 JP NZ, CLES 0066 JP 0069H 0067; 0068; 0068; WRITE SCORES TO SCREEN 0067; 0068; WRITE BOORD INPUT 0067; 0068; WRITE BOORD INPUT 0067; 0068; WRITE BOORD INPUT 0127 DEC 0128 LD (SPED), DE 0129; INCREASE SCORE 0130 LD HL, (YSCR) 0131 INC HL 0132; LD (YSCR), DE 0133; KEYBOARD SCANNING BOUTINE 0134; SCAR FOR ', 'KEY	OUAB LD (POSN), DE	OILZ INC DE
0049 LD (SPED), DR 0050; INITIAL SCORE 0051 LD DE, 0000 0052 LD (YSCR), DE 0053; CLEAR SCREEN 0056 CALL DICOH 0056 CALL FRAN 0056 CALL FRAN 0056 CALL SORT 0059; WRITE INTRODUCTION 0058; WRITE SCORES 0059; WRITE GOORES TO SCREEN 0060 CALL, FT3 0061; MAIT FOR (BOARD INPUT) 0002 CALT EYB 0063; ENTER B (C IN 'B' PRESSED) 0066 JP NZ, CLES 0067; 0068 JP OOG9H 0067; 0068; 0115; DRAN VERGE & ROAD TO LINE 1 0116 CALL DRAN 0117; 1 IN 10 CHANCE OF 0118; OBSTACLE ON BOAD 0119 LD HL, 10 0120 CALL RAND 0121 LD A, E 0123 CALL Z, OBST 0124; INCREASE SPEED 0125 LD DE, (SPED) 0126 DEC DE 0127 DEC DE 0128; INCREASE SCORE 0130 LD HL, (YSCR) 0130 LD HL, (YSCR) 0131 INC HL 0132 LD (YECR), HL 0133; REYBOARD SCANNING BOUTINE 0134; SCAN FOR ', ' KEY		OILS; SAYE BOND (OSTITUS
0050; INITIAL SCORE 0051 LD DE, 0000 0052 LD (YSCR), DE 0053; CLEAR SCREEN 0054 CALL, 01CDH 0056; WRITE INTRODUCTION 0056 CALL, FRAM 0057; SORT SCORES 0059; WRITE GOORES TO SCREEN 0059; WRITE GOORES TO SCREEN 0060 CALL, FTA 0061; WAIT FOR TBOARD INPUT 0062 CALT EYB 0063; ENTER B 1C IN 'B' PRESSED 0065 JP NZ, CLES 0124 CYSCRES 0129; INCREASE SCORE 0126 DEC 0127 DEC 0126 DEC 0127 DEC 0128 CORE 0129; INCREASE SCORE 0130 LD HL, (YSCR) 0131 INC 0132; EVECRIBED 0133; EXTROR DEC 0134; SCAR FOR 0134; SCAR FOR 0134; SCAR FOR 0134; SCAR FOR 0135; CAR FOR 0136; OSTALL DES 0137; INCREASE SCORE 0138; INCREASE SCORE 0139; INCREASE SCORE 0130 LD HL, (YSCR) 0131 INC 0132; EVECRIBED 0134; SCAR FOR 0135; CAR FOR 0135; CAR FOR 0136; CAR FOR 0137; CAR FOR 0137; CAR FOR 0138; CAR FOR 0139; CAR FOR 0139; CAR FOR 0139; CAR FOR 0130; CALL DRAW 0120; CALL BAND 0120; CALL BAND 0120; CALL BAND 0120; CALL BAND 0121; CALL DRAW 0120; CALL BAND		OLIGODAN VERGE & ROAD TO LINE 1
0051 LD DE,0000 0117;1 IN 10 CHANCE OF 0052 LD (YSCR),DE 0116;0BSTACLE ON BOAD 0053;CLEAR SCREEN 0119 LD HL,10 0054 CALL OICOH 0120 CALL RAND 0055;WRITE INTRODUCTION 0121 LD A,E 0056 CALL FRAM 0122 CP 01 0057;SORT SCURES 0123 CALL Z,OBST 0059;WRITE GOORES TO SCREEN 0124;INCREASE SPEED 0059;WRITE GOORES TO SCREEN 0125 LD DE,(EPRD) 0060 CALL,FT3 0126 DEC DE 0061;WAIT FOR (BOARD INPUT 0127 DEC DE 0063;ENTER B1C IF 'B' PRESSED 0129;INCREASE SCORE 0129;INCREASE SCORE 0130 LD HL,(YSCR) 0131 INC HL 0067; 0066 JP 0069H 0132 LD (YECR).HL 0133;EEYBOARD SCANNING BOUTINE 0125;INCREASE SCANNING BOUTINE 0125;CAN LD A, CREEN 0135;EEYBOARD SCANNING BOUTINE 0125;CAN LD A, CREEN		
0052 LD (YSCR), DE 0116; OBSTACLE ON BOAD 0053; CLEAR SCREEN 0119 LD HL, 10 0054 CALL O1CDH 0120 CALL RAND 0121 LD A, E 0055; WRITE INTRODUCTION 0121 LD A, E 0122 CP 01 0057; SORT SCURES 0123 CALL Z, OBST 0124; INCREASE SPEED 0059; WRITE SCORES TO SCREEN 0125 LD DE, (SPED) 0060 CALL, 5 TR 0061; WAIT FOR (BOARD INPUT 0127 DEC DE 0128 LD (SPED), DE 0063; ENTER B (C IF 'B' PRESSED 0130 LD HL, (YSCR) 0130 LD HL, (YSCR) 0131 INC HL 0065; DEC 0665 JP NZ, CLES 0131 INC HL 0133; REYBOARD SCANNING ROUTINE 0124; SCANNING ROUTINE 0124; SCANNING ROUTINE 0125; CAN ID A. (REFER)		0117:1 IN 10 CHANCE OF
0053; CLEAR SCREEN 0054	,	O116; OBSTACLE ON BOAD
0056 CALL FRAN 0122 CP 01 0057; SORT SCURES 0123 CALL Z,OBST 0058 CALL SORT 0124; INCREASE SPEED 0125; WRITE GOORES TO SCREEN 0125 LD DE, (SPED) 0060 CALL SORT 0126 DEC DE 0061; WAIT FOR (BOARD INPUT 0127 DEC DE 0063; ENTER B(C IM 'B' PRESSED) 0129; INCREASE SCORE 0064 CP 66 0130 LD HL, (YSCR) 0066 JP NZ, CLES 0131 INC HL 0067; 0132 LD (YECR), HL 0068; 0133; REYBOARD SCANNING BOUTINE 0125 CALL Z,OBST 0126 DEC DE 0126 DEC DE 0127 DEC DE 0128 LD (SPED), DE 0130 LD HL, (YSCR) 0131 INC HL 0132; REYBOARD SCANNING BOUTINE 0134; BCAM FOR ',' KEY	0053; CLEAR SCREEN	
0056 CALL FRAM 0057; SORT SCURES 0058 CALL SORT 0059; WRITE GOORES TO SCREEN 0060 CALL, 5 T	0054 CALI, 01C0H	
OCS7; SORT SCURES OUSS CALL SORT OUSS; WRITE SCORES TO SCREEN OUSO CALL SORT OOS1; WRITE SCORES TO SCREEN OOS2 CALL AYB OOS3; ENTER B (C IF 'B' PRESSED) OOS3; ENTER B (C IF 'B' PRESSED) OOS5 JP NZ, CLRS OOS6 JP OOS9H OOS7; OOS8; OOS8;	0055; WRITE INTRODUCTION	
0058 CALL SORT 0059; WRITE GOORES TO SCREEN 0060 CALL, 5"3 0061; WAIT FOR (BOARD INPUT 0002 CALL AYB 0063; ENTER B (C IF 'B' PRESSED) 0065 JP NZ, CLRS 0066 JP 0069H 0067; 0068;	OOSO CALL FRAM	
0059; WRITE GOORES TO SCREEN 0060		0124: INCREASE SPEED
0060 (ALJ, F-3 0061; WAIT FOR (BOARD INPUT 0002 CALT AYB 00063; ENTER B(C IN 'B' PRESSED) 0064 CP 66 0065 JP NZ, CLES 0066 JP 0069H 0067; 0068;		
0061; WAIT FOR (BOARD INPUT 0127 DEC DE 0002 CAL! AYB 0128 LD (SPED), DE 0003; ENTER B(C IN 'B' PRESSED 0129; INCREASE SCORE 0130 LD HL, (YSCR) 0131 INC HL 0065 JP NZ, CLES 0131 INC HL 0132 LD (YECR). HL 0067; 0134; BCAN FOR ', KEY 0134; BCAN FOR ', KEY	0060 CALL 5 3	O126 DEC DE
0002 CALT KYB 0003; ENTER B(C IN 'B' PRESSED) 0064 CP 66 0065 JP NZ, CLES 0066 JP 0069H 0067; 0068; 0128 LD (SPED), DE 0129; INCREASE SCORE 0130 LD HL, (YSCR) 0131 INC HL 0132 LD (YECR). HL 0133; EFYBOARD SCANNING BOUTINE 0134; SCAN FOR '.' KEY 0135; CAN LD A. (6REFE)	0061; WAIT FOR FBOARD INPUT	
UCG4 CP 66 0130 LD HL, (YSCR) 0065 JP NZ, CLES 0131 INC HL 0066 JP 0069H 0132 LD (YECR). HL 0067; 0133; KEYBOARD SCANNING BOUTINE 0268; 0134; BCAN FOR ". KEY 0135; CAN LD A. (BREFE)	0002 CALT AYB	Olza in (Seen), de
0065 JP NZ, CLES 0131 INC HL 0066 JP 0069H 0132 LD (YECR). HL 0067; 0068; 0134; SCAN FOR ', KEY 0135; CANNING MOUTINE	COMS; ENTER H (C IN 'B' PRESSED	
0066 JP 0069H 0132 LD (YECR).HL 0067; 0133; KEYBOARD SCANNING MOUTINE 0134; SCAN FOR ', KEY 0135; CAN LD A. (SREWE)	_ -	
0133; REYBOARD SCANNING MOUTINE 0134; SCAN FOR ', ' KEY 0135; CAN LD A. (BRENE)		
0068; 0134; SCAN FOR ", " KEY 0135; CAN LD A. (BRENE)	7	0133; REYBOARD SCANNING MOUTINE
A1959CAN ID A. (RREFE)		0134; SCAN FOR ', KEY
//		0135SCAN LD A, (6BEFB)
	//	

0281		14C9H	FECTO			
0283; DE =		RESULT TO INT	MACHEN .			
0284		2B058		0350; 0351	LD	DE, VIDE+452
0265	RET			0352	LD	BC, 25
0286; MOVE		TT (CARL)		0353	LDIR	
0287MVH1		DE. (CARL)		Q3 54	RET	•
0288 0289	inc LD	DE (CARL), DE		0355; SOUT	ID RXX	OTINK
0290	RET	(Chich), Mr	•	035880ND		
0291;MOVE		ř		0357 03 58	PUSH	
0292MVLE		DE, (CARL)		0358	PUSH	
0293	DEC	DE		0360	LD	HL, 0064H
0294	LD	(CART.), DE		0361	TAD .	
0295	RET	COFFI		03 68 03 68	POP	345CH
02971NSP		BL, (SPKD)		0364	POP	
0298	OK	A.		0365	POP	
0298	LD	DK, 50		0366	POP	AF
0300		RL, DE		0387	RET	
0301	LD	(SPRD), HL				R CARS ON BOAD
0302	RET	-		03690BST		П., 11
0303; DEC					CALJ,	
O304DESP		HL, (SPED)			LD ADD	HL, (POSN) HL, DE
0305	LD	DE, 50			RX	DE, HI.
0306		III., OK			LD	HL, CAR1
0307	LD	(GPED), HL			l D	BC, 2
0308 ; SOR :	RET T SAM	PES			LDIR	
0304,30%	PARR!	YOUR/CHAMPIO	N SCR		RET	
031150RT		HL, (YECR)				KEYBOARD INPUT
0312	LD	DE, (CSCH)	•	0380KEYB		ILA. BE IN 'A' RECI
0313		0V38II			PUSH	
0314	CP	OFFE			PUSH	
0315	JR	Z, NXTS		03 83		0049H
		RE IS CHAMPI	ON		POP	
0317	TD.	(CSCR), HL				DE BC
0318	KX	DE, HL PION SCORE F	LAG.		BRT	BC
0320	LD	A, 1				TRUCTIONS
0320	LD	(MERG), A		O384ERAM		HL, MSG1
0322 : COM	PARE	YOUR/SECOND	SCORE	0390		DR, VIDE+4
OJZZNXT3	1.1)	DE, (SSCR)				BC, 20
0324	CALL	Hecao			LDIR	Di verse
0325	CP	OFFH				RL, MSG2 DE, VIDE+36
0326) K	Z, 9X14				AC, 20
		RE IS SECOND	SCORE		LDIR	2.0, 2.0
0328	LD	(SSCR), HL De, Hl				HL, NSG3
0329	EX Pare	YOUR/THIRD S	CORR			DE, VIDR+66
0331NXT4		DE, (TECR)			LDIR LDIR	BC, 20
0332	CALI	Heen.				EIL, MSC4
0333	CP.	offr				DE, VIDE+100
0354	JP	z, rslt			LD	BC, 20
0335; Y 0U	r sco	RE IS THIRD	SCORE		LDIR	
0335	L.D	(TSCR), HL				BL, MSG5
0337RSLT		A, (MESG)				DE, VIDE+132
0338	CP	1 Nz, NMSG			LDIR	BC, 20
0333 0340 - 1721	JR NT (T	IVEN SCORE	:		LD	HI., MSG6
0341	LD	HL, MS16	=		LD	DE, V1DX+164
0342	LD	DE, VIDE+386	3			BC, 20
0343	LD	BC, 20			LDTR LD	HL, MSG7 ·
0344	_LD11	R	redi			DE, VIDE+196
		DEE NOT CHAMI	.70H			BC, 20 ···
03 488MS 0	i LD	HL, MS15	12		LDIR	-
			12			

0140;		0210; 0211: INCREASE SPEED
0141	JR NZ, LEFT	At 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0142	CALL MVRI	0212 LD (SPED), HL 0213; INCREASE MINIMUM SPEED
0143	JR SPCT	0214 LD DE, 100
0144;362	M FOR 'M' KEY	
	CP ODFH	O215 BBC (IL, DE O216 LD (MINS), BL
014R		
0147	CHRIST EX (ALE)	0217; KEEP SPEED DELAY >208
)	N NOR 'Q' KEY	0218SPC3 LD IIL, (SPED)
014920°C	LD A. (68FEH) · CP OEFH	0219 LD DE, 20
		O220 CALL OA39H
0101	JR NZ, BRKE CALL INSP	0221 CP 01
0152 0153	TH MHOA	0222 JR Z,DLAY
U1 DO D1 E A + C C A	M ROB 'A' KEA	0223 LD (SPED), DE
	LD A, (68FDII)	0224; SPINED CONTROL DELAY
0155 DR A	CP OFFH	OZZEDLAY LD BC, (SPRD)
	JR NZ, NMOV	0226 CALL 0080H
0158	CALL DESP	0227; MAKE NOISES
0159 : GET	LOCATION OF ARKA	0228 CALL SOND
	FRONT OF CAR	0229; CONTINUE GAME
DIR INMOV	LD HL, (CARL)	0230 JP SCRL
	LD DF, 82	0231; SIMULATE CAR CRASH
	SBC HL, DE	0232; Flash Cars 50 Times
	CK IF BOAD IS CLEAR	0233CRSH LD B, 50
0185CHK1	LD A, (HL)	02341AP6 BXX
0166	CP OBFIL	0235 LD HL, Carr
	JR Z, Clik2	0236 LD DE, (CARL)
3168	CP OB7H	0237 LD BC, 2
0169	JR Z, CHK?	6238 HALT
170:SKT	CAR CRASHED FLAG	O238 LDIR
0171	J.D A, 01	0240 LD BC, 2000
172	LD (FLAG), A	0241 CALL 0060H
0173	JR DCAR	O242 CALL DRCR
)174CHK2	INC HL	0243 LD BC, 2000
175	LD A, (HL)	0244 CALL 0080H
176	CP OBFH	0245 KXX
0177	JR Z, DCAR	024G DJNZ LOP5
)178	CP OB7H	0247 LD BC, 1-2
0179	JR Z, DCAR	0248 CALL 0060H
0180; SET	CAR CRASHED FLAG	0249 XOR A
0181	LD A, 01	0250 LD (FLAG), A
29162	LD (FLAG), A	0251; WRITE NEW SCORE TO SCREEN
0189; DRA	M CAR	0252 CALL 01C9H
0184DCAE	CALL DRCR	0283 LD HL, MS14
0185:DII	CAR CRASH	0254 LD DE, VIDE+100
0186	LD A, (FLAG)	0255 LD BC, 13
0187	CP 01	0256 LDIR
0168	JR Z, CRSH	0257 CALL 0A9DH
0189; MAP	E GAME MORE DIFFICULT	0258 LD HL, YSCR
0190:AS	SCORE THURRASES	
0191; CO	PARE SCORE TO MAX SCORE	
D1825PC1	LD DE, (YSCR)	0260 CALJ, OFBDH 0261 LD DK, VIDK+115
0193	LD HL, (MSCR)	0262 LD (7820H), DE
0194	CULT OV38H	0263 CALL 2B75H
0195	CP 01	0264; SOR'T SCORES
0196	.ir z.stc3	
0197; I H 0	CREASE MAXIMUM SCORE	0265 CALL SORT 0266; RESET CHAMPION SCORE FLAC
0198	LD DE, 100	
03.91)	ADD BI., DR	
osoo	IJ) (MSCB), FIL	Q768 LD (MESG), A
	PARE SPEED TO MIN SPEED	0269; WAIT FOR KRYBOARD INPUT
	LD DE, (SI'RD)	0270 CALL KEYB
0203	LD BL, (MINS)	0271 JP STRT
0204	CALL QASOFI	0272; GET A RANDOM NUMBER
0205	CP OFFH	0273; RETURN INTEGER VALUE IN
0206	-3r NZ, SPC3	0274; TO WRA1 = ACC = 79218
0207; 0208;	• ,	OZ75RAND CALL OA9AH OZ75;GENERATE A BANDOM NUMBER
		A TABLE OF THE PERSON OF THE P

		,				
0417	LD HL, MSG8		491 492 ; DRAV	LDIR		
.0418	LD DE, VIDE+228				HL. ROAD	
0419	LT BC; 22	Ŏ	494	1.0	DE, (POSK)	
0420	LDIA	ō	495		BC, 18	
9428	LD HL, MSG9			LDIR		•
0428	LD DE. VIDE+260	0		RET		
0427	LD BC, 22	C	490; COPT	LIN	C 1 TO MUCL	e screin
042R	LDIR	9	499COPY	TD.	HL, VIDE	. •
0429	LD HI, MB10		500 501	LD LD	DE, VIDE+32 BC, 448	
0430	ID DE VIDE+292			HALT	DC, 440	
0431 0432	LD BC, 22 LDIR	•	509	LDIR		
0432	LD HI, MS11	Ò		RET		
434	LD DE, VIDR+324	O	805; DRAF			
ે43&	LD BC, 22		506DRCR	LD	HL, CAAR	
0436	LDIR				DE, (CARL)	
0437	LD ILL, MS12			LD HALT	BC, 2	
0438	LD DR. VIDE+356			LDIR	•	
0439	LD BC, 22	_		RET		
0440	LDIR		512; STA			
0441	RET		513VIDE			
0442, 12	RINT SCORES RINT CHAMPION SCORE RT NTY FLAG=2 (INTEGER)		614; CHAL	DEED BYON	PLAG	
DAAA-G)	OF NTF FLAG=2 (INTEGER)		CAD SIN	DECI	ENI	
0.127,100	OR CALL OASDH	ř	K17CAAR	DERW	ለዓ79ጽዚ	
0446	LD HI, CECR	č	518:0BS1	PACIJE	CAR DESIGN	
	OAD 4 BYTES FROM THE	- 7	519CAR1	DEFW	09996U	•
	CATION POINTED TO BY EL	•	520; CRA	SH CAI	O O O O O O O O O O O O O O O O O O O	
	TO THE ACCUMULATOR (ACC)	Ç	521CAR2	DENH	O9D9RH	
0450	CALL OUBIH	. 9	522; CAR	LOCA	TION ON SCH	EEN
	MVERTS ACC TO ASC11		1623CAKL 1524 ; СНА)		VIDE+496	
	TRING DELIMITED BY ZERO		525CSCR			
0453	CALL OFBDH		526	DEFW		•
0454	LD DE, VIDE+306	, ()527 ; SEC (Ba CIMC	CORE	
0455; C	JRSOR LOCATION LD (78208), DE		062855CR			
	RINT MESSAGE POINTED TO	()529 	DEFW		
	HL. MUST END IN ZERO.)530; Thi)531 TSCR			
0459	CALL 2B75H			DEAM		
	RINT CARRIAGE RETURN		0533; YOU			
0461	LD A, ODH		534Y5CR			
0482	CALL OSBAH		0636	DEFR	0	
0463;PI	RINT SECOND SCORE				R SPEED CON	TROL.
0484	CALL OASDH	4	D5376PED		SCORE FLAG	
0485	LD HL, SSCR Call 098111		0539MSCR			
0466 0467	CALL OFBDH	1	0540; MIH	IMUM	SPRED FLAG	•
0468	LD DE, VIDE+340		0541 min s			_
0469	LD (7820H), DE		0542;ROA	D POS	ITION ON 80	
0470	CALL 2B75H	•	0543 <u>PUSN</u> 0544 - CAP	PAG	HED FLAG	
0471;Pl	RINT CARRIAGE RETURN	•	0545FLAG	DEFR	O PLESS	
0472	LD A, ODH		0546; ROA	d des	IGN	
0473	CALL 033AH		OS 47 BOAD	DEFU	OBFB7H	-
	RINT THIRD SCORE		0548		OBTBFH	
0475	Call Oa9dh Ld Hl, Tscr		054 9 0550	DEFA	OBFBFH OBFBFH	
047 6 047 7	CALL O9B1H		0551		OBFB7H	
0477	CALL OFBOH		0552	DEFA	OBTEVE	
0479	LD DE, VIDE+372		0553		OBFBFH	
0480	LD (7820H), DE		0554		OBERTH	٠
0481	CALL 2B75H	:	0555; MRS 0556MSQ1	PANT	• •	•
0482	HET	•	0557;		•	60TO PAGE 9
	RAW VERGE		0558			UVIU INUL U
0484UR		1.4	-	•	•	
0485	LD DE, VIDE	14				
0486	LD BC, 32					
0487; 0488;						
UTUU,						